

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	244	717/128.ccls.	USPAT	OR	OFF	2006/01/25 11:45
L2	0	717/128.ccls. and (compress\$3 and uncompress\$3)	USPAT	OR	OFF	2006/01/25 11:46
L3	22	717/128.ccls. and (compress\$3)	USPAT	OR	OFF	2006/01/25 11:46
L4	15	717/128.ccls. and (compress\$3) and (branch\$3 and start\$3)	USPAT	OR	OFF	2006/01/25 11:49
L5	9	(compress\$3 and uncompress\$3) and (branch\$3 and start\$3) same (trace or tracing)	USPAT	OR	OFF	2006/01/25 11:54
L6	9	714/45.ccls. and (compress\$3 and uncompress\$3)	USPAT	OR	OFF	2006/01/25 11:55
S1	5	generat\$3 adj compress\$3 same (instruction adj address)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/04/08 13:00
S2	25	"5632024".URPN.	USPAT	OR	OFF	2004/04/08 13:01
S3	25	"5632024".URPN.	USPAT	OR	OFF	2004/04/08 13:57
S4	5	"5632024".URPN. and (branch\$3 same compress\$3 and (event or match\$3))	USPAT	OR	OFF	2004/04/08 14:02
S5	0	"5632024".URPN. and compress\$3 same (event near3 trac\$3)	USPAT	OR	OFF	2004/04/08 14:04
S6	0	"5632024".URPN. and (event near3 trac\$3)	USPAT	OR	OFF	2004/04/08 14:04
S7	132	((detect\$3 near3 event) near5 trac\$3)	USPAT	OR	OFF	2004/04/08 14:05
S8	99	((detect\$3 near3 event) near3 trac\$3)	USPAT	OR	OFF	2004/04/08 14:12
S9	14	((detect\$3 near3 event) near3 trac\$3) and compress\$3	USPAT	OR	OFF	2004/04/08 14:05
S10	0	((detect\$3 near3 event) near3 trac\$3) and latch adj signal	USPAT	OR	OFF	2004/04/08 14:13
S11	10	((detect\$3 near3 event) near3 trac\$3) and (signal near3 active same (timeframe or time-frame or time or period))	USPAT	OR	OFF	2004/04/08 14:16
S12	0	(event adj manag\$4) same (signal adj active)	USPAT	OR	OFF	2004/04/08 14:17
S13	227	(event adj manag\$4) same (time or time-frame or (time adj frame) or period)	USPAT	OR	OFF	2004/04/08 14:17

S14	0	(event adj manag\$4) same (time or time-frame or (time adj frame) or period) and data adj latch	USPAT	OR	OFF	2004/04/08 14:17
S15	12	(event adj manag\$4) and data adj latch	USPAT	OR	OFF	2004/04/08 14:18
S16	46	"5535331".URPN.	USPAT	OR	OFF	2004/04/08 14:20
S17	7	"5535331".URPN. and compress\$4	USPAT	OR	OFF	2004/04/08 14:26
S18	5	"5535331".URPN. and compress\$4 and branch\$3	USPAT	OR	OFF	2004/04/08 14:26
S19	7	("4641348" "5535331" "5632024" "5764885" "5764994" "5784585" "6216213").pn.	USPAT	OR	OFF	2004/04/12 15:14
S20	168	717/128.ccls.	USPAT	OR	OFF	2004/04/12 16:03
S21	14	717/128.ccls. and compress\$3	USPAT	OR	OFF	2004/04/12 16:03
S22	2276	(trac\$3 or instrument\$5 or profil\$3) same (compress\$3 or "reduce file size" or "reduce data size" or "reduction in data" or "reduction in file") and (uncompress\$3 or "not compressing" or "not compressed" or "not reduce" or "no reduction" or "no compression")	USPAT	OR	OFF	2005/03/21 16:06
S23	557	(trac\$3 or profil\$3) same ((compress\$3 or "reduce file size" or "reduce data size" or "reduction in data" or "reduction in file") and (uncompress\$3 or "not compressing" or "not compressed" or "not reduce" or "no reduction" or "no compression"))	USPAT	OR	OFF	2005/03/21 16:06
S24	353	(trac\$3) same ((compress\$3 or "reduce file size" or "reduce data size" or "reduction in data" or "reduction in file") and (uncompress\$3 or "not compressing" or "not compressed" or "not reduce" or "no reduction" or "no compression"))	USPAT	OR	OFF	2005/03/21 16:13
S25	6	(trace and compress\$3).ti.	USPAT	OR	OFF	2005/03/21 16:07
S26	22	(trac\$3 and compress\$3).ti.	USPAT	OR	OFF	2005/03/21 16:07
S27	3	("5327361").PN. OR ("6087967").URPN.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/21 16:10

S28	305	(trac\$3) same ((compress\$3 or "reduce file size" or "reduce data size" or "reduction in data" or "reduction in file") same (uncompress\$3 or "not compressing" or "not compressed" or "not reduce" or "no reduction" or "no compression"))	USPAT	OR	OFF	2005/03/21 16:16
S29	117	(trac\$3) same ((compress\$3 or "reduce file size" or "reduce data size" or "reduction in data" or "reduction in file") same (uncompress\$3 or "not compressing" or "not compressed" or "not reduce" or "no reduction" or "no compression")) and (branch\$3 or loop\$3)	USPAT	OR	OFF	2005/03/21 16:14
S30	42	(trace or traced or tracing) same ((compress\$3 or "reduce file size" or "reduce data size" or "reduction in data" or "reduction in file") same (uncompress\$3 or "not compressing" or "not compressed" or "not reduce" or "no reduction" or "no compression"))	USPAT	OR	OFF	2005/03/21 16:17
S31	9	("5471593" "5651137" "5826101" "5950009" "6012052" "6212629" "6272649" "6298370").PN. OR ("6542855").URPN.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/21 16:21
S32	9	("5471593" "5651137" "5826101" "5950009" "6012052" "6212629" "6272649" "6298370").PN. OR ("6542855").URPN. and ((compress\$3 or "reduce file size" or "reduce data size" or "reduction in data" or "reduction in file") same (uncompress\$3 or "not compressing" or "not compressed" or "not reduce" or "no reduction" or "no compression"))	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/21 16:25
S33	9	("5471593" "5651137" "5826101" "5950009" "6012052" "6212629" "6272649" "6298370").PN. OR ("6542855").URPN. and ((compress\$3 or "reduce file size" or "reduce data size" or "reduction in data" or "reduction in file") same (uncompress\$3 or "not compressing" or "not compressed" or "not reduce" or "no reduction" or "no compression"))	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/21 16:26

S34	1	((("5471593" "5651137" "5826101" "5950009" "6012052" "6212629" "6272649" "6298370").PN. OR ("6542855").URPN.) and ((compress\$3 or "reduce file size" or "reduce data size" or "reduction in data" or "reduction in file") same (uncompress\$3 or "not compressing" or "not compressed" or "not reduce" or "no reduction" or "no compression"))	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/21 16:27
S35	6	("20020184477" "6094729" "6243836" "6314530" "6347383" "6542855").PN. OR ("6615371").URPN.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/21 16:29
S36	2	((("20020184477" "6094729" "6243836" "6314530" "6347383" "6542855").PN. OR ("6615371").URPN.) and ((compress\$3 or "reduce file size" or "reduce data size" or "reduction in data" or "reduction in file") same (uncompress\$3 or "not compressing" or "not compressed" or "not reduce" or "no reduction" or "no compression"))	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/21 16:29

S37	84	("4814981" "5251311" "5355487" "5386565" "5423050" "5434804" "5440705" "5448576" "5452432" "5455936" "5479652" "5483518" "5488688" "5530965" "5570375" "5590354" "5596734" "5598551" "5608881" "5613153" "5627842" "5657273" "5659679" "5682545" "5704034" "5708773" "5724505" "5724549" "5737516" "5751621" "5768152" "5771240" "5774701" "5778237" "5781558" "5796978" "5809293" "5828825" "5832248" "5835963" "5848247" "5848264" "5860127" "5862387" "5867726" "5884092" "5896550" "5918045" "5930523" "5930833" "5938778" "5943498" "5944841" "5950012" "5953538" "5956477" "5978874" "5978902" "5983017" "5983366" "5983379" "5996092" "5999112" "6094729" "6108761" "6145099" "6145123" "6148381" "6154857" "6167499" "6185732" "6243836" "6269454" "6282701" "6314530" "6345295" "6370660").PN. OR ("6615370"). URPN.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/21 17:28
-----	----	--	------------------------------	----	-----	------------------

S38	0	("4814981" "5251311" "5355487" "5386565" "5423050" "5434804" "5440705" "5448576" "5452432" "5455936" "5479652" "5483518" "5488688" "5530965" "5570375" "5590354" "5596734" "5598551" "5608881" "5613153" "5627842" "5657273" "5659679" "5682545" "5704034" "5708773" "5724505" "5724549" "5737516" "5751621" "5768152" "5771240" "5774701" "5778237" "5781558" "5796978" "5809293" "5828825" "5832248" "5835963" "5848247" "5848264" "5860127" "5862387" "5867726" "5884092" "5896550" "5918045" "5930523" "5930833" "5938778" "5943498" "5944841" "5950012" "5953538" "5956477" "5978874" "5978902" "5983017" "5983366" "5983379" "5996092" "5999112" "6094729" "6108761" "6145099" "6145123" "6148381" "6154857" "6167499" "6185732" "6243836" "6269454" "6282701" "6314530" "6345295" "6370660").PN. OR ("6615370"). URPN.) and ((compress\$3 or "reduce file size" or "reduce data size" or "reduction in data" or "reduction in file") same (uncompress\$3 or "not compressing" or "not compressed" or "not reduce" or "no reduction" or "no compression"))	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/21 16:42
-----	---	--	------------------------------	----	-----	------------------

S39	0	("4814981" "5251311" "5355487" "5386565" "5423050" "5434804" "5440705" "5448576" "5452432" "5455936" "5479652" "5483518" "5488688" "5530965" "5570375" "5590354" "5596734" "5598551" "5608881" "5613153" "5627842" "5657273" "5659679" "5682545" "5704034" "5708773" "5724505" "5724549" "5737516" "5751621" "5768152" "5771240" "5774701" "5778237" "5781558" "5796978" "5809293" "5828825" "5832248" "5835963" "5848247" "5848264" "5860127" "5862387" "5867726" "5884092" "5896550" "5918045" "5930523" "5930833" "5938778" "5943498" "5944841" "5950012" "5953538" "5956477" "5978874" "5978902" "5983017" "5983366" "5983379" "5996092" "5999112" "6094729" "6108761" "6145099" "6145123" "6148381" "6154857" "6167499" "6185732" "6243836" "6269454" "6282701" "6314530" "6345295" "6370660").PN. OR ("6615370"). URPN.) and ((compress\$3 or "reduce file size" or "reduce data size" or "reduction in data" or "reduction in file") and (uncompress\$3 or "not compressing" or "not compressed" or "not reduce" or "no reduction" or "no compression"))	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/21 16:42
-----	---	--	------------------------------	----	-----	------------------

S40	0	(("4814981" "5251311" "5355487" "5386565" "5423050" "5434804" "5440705" "5448576" "5452432" "5455936" "5479652" "5483518" "5488688" "5530965" "5570375" "5590354" "5596734" "5598551" "5608881" "5613153" "5627842" "5657273" "5659679" "5682545" "5704034" "5708773" "5724505" "5724549" "5737516" "5751621" "5768152" "5771240" "5774701" "5778237" "5781558" "5796978" "5809293" "5828825" "5832248" "5835963" "5848247" "5848264" "5860127" "5862387" "5867726" "5884092" "5896550" "5918045" "5930523" "5930833" "5938778" "5943498" "5944841" "5950012" "5953538" "5956477" "5978874" "5978902" "5983017" "5983366" "5983379" "5996092" "5999112" "6094729" "6108761" "6145099" "6145123" "6148381" "6154857" "6167499" "6185732" "6243836" "6269454" "6282701" "6314530" "6345295" "6370660").PN. OR ("6615370"). URPN.) and ((compress\$3 or "reduce file size" or "reduce data size" or "reduction in data" or "reduction in file") same (uncompress\$3 or "not compressing" or "not compressed" or "not reduce" or "no reduction" or "no compression"))	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/21 17:29
S41	9	("5809293").URPN.	USPAT	OR	OFF	2005/03/22 07:55
S42	42	(trac\$3) same (compress\$3 near2 scheme)	USPAT	OR	OFF	2005/03/22 07:59
S43	7	(trac\$3) near5 branch\$3 near5 compress\$3	USPAT	OR	OFF	2005/03/22 07:59
S44	7	(trac\$3) near5 branch\$3 near5 compress\$3	USPAT	OR	OFF	2005/03/22 08:00

S45	1	(trace or tracing or traced) near5 branch\$3 near5 compress\$3	USPAT	OR	OFF	2005/03/22 08:06
S46	1	"5764994".pn.	USPAT	OR	OFF	2005/03/22 08:06
S47	23	("4355306" "4410916" "4558302" "4862167" "4876541" "5126739" "5146221" "5155484" "5319793" "5353061" "5369605" "5398319" "5414425" "5414833" "5463390" "5632024" "5652852").PN. OR ("5764994"). URPN.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/22 08:33
S48	19070	determin\$3 near3 compress\$3	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/22 08:34
S49	0	"determin\$3 when to compress"	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/22 08:34
S50	0	"determin\$3 when" near2 compress\$3	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/22 08:35
S51	0	"determine when" near2 compress\$3	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/22 08:35
S52	0	"determining when" near2 compress\$3	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/22 08:35
S53	46	"trace data" near2 compress\$3	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/22 08:36
S54	42	"trace data" near2 compress\$3 and (branch\$3 or match\$3 or compar\$4)	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/22 08:37
S55	14	"trace data" near2 compress\$3 and "instruction address"	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/22 08:38

S56	83	("3707725" "4429368" "4462077" "4598364" "4611281" "5058114" "5321828" "5345580" "5357626" "5371689" "5394544" "5446876" "5469571" "5488688" "5491793" "5526485" "5530804" "5533192" "5544311" "5594903" "5615331" "5630102" "5630128" "5634046" "5642479" "5678003" "5689694" "5724505" "5740440" "5751942" "5752013" "5764885" "5768152" "5771240" "5774708" "5802272" "5812811" "5828824" "5848264" "5867644" "5889981" "5889988" "5901283" "5903718" "5943498" "5953530" "5978902" "5978937" "5996092" "6009270" "6041406").PN. OR ("6314530"). URPN.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/22 08:48
S57	8	(("3707725" "4429368" "4462077" "4598364" "4611281" "5058114" "5321828" "5345580" "5357626" "5371689" "5394544" "5446876" "5469571" "5488688" "5491793" "5526485" "5530804" "5533192" "5544311" "5594903" "5615331" "5630102" "5630128" "5634046" "5642479" "5678003" "5689694" "5724505" "5740440" "5751942" "5752013" "5764885" "5768152" "5771240" "5774708" "5802272" "5812811" "5828824" "5848264" "5867644" "5889981" "5889988" "5901283" "5903718" "5943498" "5953530" "5978902" "5978937" "5996092" "6009270" "6041406").PN. OR ("6314530"). URPN.) and "instruction address" and compress\$4	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/22 09:54

S58	2	("5535331" "5764994").pn.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/22 10:10
S59	1	("5491793").pn.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/03/22 10:10
S60	44	("5491793").URPN.	USPAT	OR	OFF	2005/03/22 10:12
S61	41	("5491793").URPN. and (stor\$3 or compress\$4 or decompress\$4 or ("not" and compress\$4) or ("no" adj compress\$4))	USPAT	OR	OFF	2005/03/22 10:15
S62	9	("5491793").URPN. and (stor\$3 or writ\$3) same (compress\$4 or decompress\$4 or ("not" and compress\$4) or ("no" adj compress\$4))	USPAT	OR	OFF	2005/03/22 10:19
S63	444	austin.in. and trac\$3	USPAT	OR	OFF	2005/03/22 10:19
S64	3	austin.in. and trac\$3 and (detect\$3 near2 event)	USPAT	OR	OFF	2005/03/22 10:20
S65	238	hohl.in.	USPAT	OR	OFF	2005/03/22 10:21
S66	0	hohl.in. and (emulation near3 circuit)	USPAT	OR	OFF	2005/03/22 10:21
S67	6	hohl.in. and emulat\$3	USPAT	OR	OFF	2005/03/22 10:22
S68	0	hohl.in. and emulat\$3 and (stor\$3 or writ\$3) same (compress\$4 or decompress\$4 or ("not" and compress\$4) or ("no" adj compress\$4))	USPAT	OR	OFF	2005/03/22 10:23
S69	716544	hohl.in. and emulat\$3 and (stor\$3 or writ\$3) or (compress\$4 or decompress\$4 or ("not" and compress\$4) or ("no" adj compress\$4))	USPAT	OR	OFF	2005/03/22 10:23
S70	6	hohl.in. and emulat\$3 and ((stor\$3 or writ\$3) or (compress\$4 or decompress\$4 or ("not" and compress\$4) or ("no" adj compress\$4)))	USPAT	OR	OFF	2005/03/22 10:25
S71	3	("5491793" "5740413" "5621886").pn.	USPAT	OR	OFF	2005/03/22 11:10
S72	1	("5535331").pn. and trac\$3	USPAT	OR	OFF	2005/03/22 11:10
S73	9	(trace or tracing or monitor\$3 or debug\$4) near2 (portion or section or segment) same (compress\$3) and (start near3 (instruction or address or location))	USPAT	OR	OFF	2005/08/04 08:07

S74	6	("5754827" "6094729" "6145100" "6243836" "6314530" "6594185").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/04 08:06
S75	14342	(trace or tracing or monitor\$3 or debug\$4) and (decompress\$3 or reconstruct\$3 or construct\$3 or calculat\$3 or obtain\$3) near3 (initial or start or first or begining or front) and (compress\$3) and (instruction or address or location or section or segment or portion)	USPAT	OR	OFF	2005/08/04 08:09
S76	1025	(trace or tracing or monitor\$3 or debug\$4) same (decompress\$3 or reconstruct\$3 or construct\$3 or calculat\$3 or obtain\$3) near3 (initial or start or first or begining or front) and (compress\$3) and (instruction or address or location or section or segment or portion)	USPAT	OR	OFF	2005/08/04 08:10
S77	303	(trace or tracing or monitor\$3 or debug\$4) same (decompress\$3 or reconstruct\$3 or construct\$3) near3 (initial or start or first or begining or front) and (compress\$3) and (instruction or address or location or section or segment or portion)	USPAT	OR	OFF	2005/08/04 08:10
S78	68	(trace or tracing or monitor\$3 or debug\$4) same (decompress\$3 or reconstruct\$3 or construct\$3) near3 (initial or start or first or begining or front) same (instruction or address or location or section or segment or portion) and (compress\$3)	USPAT	OR	OFF	2005/08/04 08:27
S79	14	("5752014" "5784604" "5805877" "5864697" "5978902" "6009270" "6081887" "6088793" "6094729" "6112293").PN. OR ("6247146").URPN.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/04 08:19
S80	14	("5752014" "5784604" "5805877" "5864697" "5978902" "6009270" "6081887" "6088793" "6094729" "6112293").PN. OR ("6247146").URPN.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/08/04 08:27

S81	16244	(trace or tracing or monitor\$3 or debug\$4) same (uncompress\$3 or reconstruct\$3 or decod\$3 or read\$3) and (initial or start or first or begining or front) same (instruction or address or location or section or segment or portion) and (compress\$3)	USPAT	OR	OFF	2005/08/04 08:28
S82	984	(trace or tracing or monitor\$3 or debug\$4) same (uncompress\$3 or reconstruct\$3) and (initial or start or first or begining or front) same (instruction or address or location or section or segment or portion) and (compress\$3)	USPAT	OR	OFF	2005/08/04 08:29
S83	84	(trace or tracing or monitor\$3 or debug\$4) same (uncompress\$3 or reconstruct\$3) same (initial or start or first or begining or front) same (instruction or address or location or section or segment or portion) and (compress\$3)	USPAT	OR	OFF	2005/08/04 08:29
S84	22	("5764885").URPN.	USPAT	OR	OFF	2005/08/04 08:40
S85	22	("6145123").URPN.	USPAT	OR	OFF	2005/08/04 09:18
S86	0	(compress\$3 adj ("all but" or "all except for")) near3 (initial or start\$3 or first or begining or front)	USPAT	OR	OFF	2005/08/04 09:20
S87	1	("do not" or " don't") near3 (compress\$3) near5 (initial or start\$3 or first or begining or front)	USPAT	OR	OFF	2005/08/04 09:35
S88	20	"instruction address trace"	USPAT	OR	OFF	2005/08/04 09:38
S89	5	nec.as. and (tracer or trace) and (compress\$3) and (branch\$2)	USPAT	OR	OFF	2005/08/04 11:51
S90	0	omron.as. and (tracer or trace) and (compress\$3) and (branch\$2)	USPAT	OR	OFF	2005/08/04 11:51
S91	110	shimizu.in. and masaru.in.	USPAT	OR	OFF	2005/08/04 11:51
S92	0	(shimizu.in. and masaru.in.) and (trace or traced or tracing) and compress\$3	USPAT	OR	OFF	2005/08/04 11:52
S93	7	(shimizu.in. and masaru.in.) and (trace or traced or tracing)	USPAT	OR	OFF	2005/08/04 11:52
S94	19	(shimizu.in. and masaru.in.) and compress\$3	USPAT	OR	OFF	2005/08/04 11:52
S95	216	partial near3 (trace or tracing)	USPAT	OR	OFF	2005/09/07 08:53
S96	9	partial near3 (trace or tracing) same:compress\$3	USPAT	OR	OFF	2005/09/07 11:10

S97	35	selective near2 compress\$3 and (trace or tracing)	USPAT	OR	OFF	2005/09/07 11:11
S98	17	selective near2 compress\$3 and (trace or tracing) and (instruction or address or branch or conditional)	USPAT	OR	OFF	2005/09/07 11:15
S99	0	"do not compress branch" or "without compressing branch" or "do not compress conditional" or "without compressing conditional"	USPAT	OR	OFF	2005/09/07 11:25
S100	17	"do not compress branch" or "without compressing branch" or "do not compress conditional" or "without compressing conditional" or uncompressed near2 (conditional or branch\$3)	USPAT	OR	OFF	2005/09/07 11:26
S101	23	("4355306" "4410916" "4558302" "4862167" "4876541" "5126739" "5146221" "5155484" "5319793" "5353061" "5369605" "5398319" "5414425" "5414833" "5463390" "5632024" "5652852").PN. OR ("5764994").URPN.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/09/07 12:42
S102	1	write\$3 near2 address near2 uncompressed	US-PGPUB; USPAT; USOCR	OR	OFF	2005/09/07 12:43
S103	1	branch\$3 near2 address near2 uncompressed	US-PGPUB; USPAT; USOCR	OR	OFF	2005/09/07 14:47
S104	8	("6145123" "6247146" "6094729" "5764885" "5802272" "6314530" "6684348" "5943498").pn.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/09/07 15:32
S105	8	("6145123" "6247146" "6094729" "5764885" "5802272" "6314530" "6684348" "5943498").pn. AND (START\$3 OR BEGIN\$3 OR ENTRY OR FRESH OR BOUNDARY OR WORD OR FIRST)	US-PGPUB; USPAT; USOCR	OR	OFF	2005/09/07 15:59
S106	6	("6145123" "6247146" "6094729" "5764885" "5802272" "6314530" "6684348" "5943498").pn. AND (START\$3 OR BEGIN\$3 OR ENTRY OR FRESH OR BOUNDARY OR WORD OR FIRST) AND COMPRESS\$3	US-PGPUB; USPAT; USOCR	OR	OFF	2005/09/07 16:23

S10 7	14	(TRACE OR TRACED OR TRACING) same (stor\$3 or record\$3 or log\$4 or journal\$4) near4 uncompressed	US-PGPUB; USPAT; USOCR	OR	OFF	2005/09/07 16:24
----------	----	--	------------------------------	----	-----	------------------



USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Log out](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Published since January 1980 and Published before March 2001

Found 987 of 97

Terms used **compress trace conditional**

Sort results

by

☒ [Save results to a Binder](#)
[Try an Advanced Search](#)
☒ [Search Tips](#)
[Try this search in The ACM Guide](#)

Display results

☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☒

1 [Deterministic delay bounds for VBR video in packet-switching networks: fundamental limits and practical trade-offs](#)

Dallas E. Wrege, Edward W. Knightly, Hui Zhang, Jörg Liebeherr

June 1996 **IEEE/ACM Transactions on Networking (TON)**, Volume 4 Issue 3**Publisher:** IEEE PressFull text available: ☒ pdf(1.51 MB)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)

2 [Techniques for compressing program address traces](#)

☒ Andrew R. Pleszkun
November 1994 **Proceedings of the 27th annual international symposium on Microarchitecture****Publisher:** ACM PressFull text available: ☒ pdf(931.63 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


In this paper a technique for generating consistent, reproducible traces with about an order of magnitude better compression than standard general-purpose compression programs is described. With this approach, the trace is read once, an intermediate form is generated and then read as the input to the second pass over the address stream. No program source code is required and this technique will work on address streams that include OS calls. As a result of the way the address trace is encoded ...

Keywords: compression, trace generation

3 [Fundamental limits and tradeoffs of providing deterministic guarantees to VBR video traffic](#)

☒ Edward W. Knightly, Dallas E. Wrege, Jörg Liebeherr, Hui Zhang
May 1995 **ACM SIGMETRICS Performance Evaluation Review**, **Proceedings of the 1995 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems SIGMETRICS '95/PERFORMANCE '95**, Volume 23 Issue 1

Publisher: ACM Press

Full text available:  [pdf\(1.08 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Compressed digital video is one of the most important traffic types in future integrated services networks. However, a network service that supports delay-sensitive video imposes many problems since compressed video sources are variable bit rate (VBR) with a high degree of burstiness. In this paper, we consider a network service that can provide deterministic guarantees on the minimum throughput and the maximum delay of VBR video traffic. A common belief is that due to the burstiness of VBR traf ...


4 [RecPlay: a fully integrated practical record/replay system](#)

Michiel Ronsse, Koen De Bosschere



May 1999 **ACM Transactions on Computer Systems (TOCS)**, Volume 17 Issue 2

Publisher: ACM Press

Full text available:  [pdf\(324.00 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This article presents a practical solution for the cyclic debugging of nondeterministic parallel programs. The solution consists of a combination of record/replay with automatic on-the-fly data race detection. This combination enables us to limit the record phase to the more efficient recording of the synchronization operations, while deferring the time-consuming data race detection to the replay phase. As the record phase is highly efficient, there is no need to switch it off, hereby elimi ...



Keywords: binary code modification, multithreaded programming, race detection

5 [Path-based next trace prediction](#)

Quinn Jacobson, Eric Rotenberg, James E. Smith

December 1997 **Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture**

Publisher: IEEE Computer Society

Full text available:  [pdf\(1.15 MB\)](#) 

[Publisher Site](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The trace cache has been proposed as a mechanism for providing increased fetch bandwidth by allowing the processor to fetch across multiple branches in a single cycle. But to date predicting multiple branches per cycle has meant paying a penalty in prediction accuracy. We propose a next trace predictor that treats the traces as basic units and explicitly predicts sequences of traces. The predictor collects histories of trace sequences (paths) and makes predictions based on these histories. The b ...


Keywords: Trace Cache, Next Trace Prediction, Multiple Branch Prediction, Return History Stack, Path-Based Prediction

6 [Two-way TCP traffic over rate controlled channels: effects and analysis](#)

Lampros Kalampoukas, Anujan Varma, K. K. Ramakrishnan

December 1998 **IEEE/ACM Transactions on Networking (TON)**, Volume 6 Issue 6


Publisher: IEEE Press

Full text available:  pdf(393.35 KB)

Additional Information: [full citation](#), [references](#), [index terms](#)


Keywords: TCP over ATM, congestion control, rate control, two-way traffic

7 [Optimally profiling and tracing programs](#)

 Thomas Ball, James R. Larus

July 1994 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 16 Issue 4

Publisher: ACM Press


Full text available:  pdf(2.84 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This paper describes algorithms for inserting monitoring code to profile and trace programs. These algorithms greatly reduce the cost of measuring programs with respect to the commonly used technique of placing code in each basic block. Program profiling counts the number of times each basic block in a program executes. Instruction tracing records the sequence of basic blocks traversed in a program execution. The algorithms optimize the placement of counting/tracing code with respect to the ...


Keywords: control-flow graph, instruction tracing, instrumentation, profiling

8 [Completeness and incompleteness of trace-based network proof systems](#)

 J. Widom, D. Gries, F. B. Schneider

October 1987 **Proceedings of the 14th ACM SIGACT-SIGPLAN symposium on Principles of programming languages**


Publisher: ACM Press

Full text available:  pdf(1.30 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


Most trace-based proof systems for networks of processes are known to be incomplete. Extensions to achieve completeness are generally complicated and cumbersome. In this paper, a simple trace logic is defined and two examples are presented to show its inherent incompleteness. Surprisingly, both examples consist of only one process, indicating that network composition is not a cause of incompleteness. Axioms necessary and sufficient for the relative completeness of a trace logic are then pre ...

9 [Fast and accurate instruction fetch and branch prediction](#)

 B. Calder, D. Grunwald

April 1994 **ACM SIGARCH Computer Architecture News , Proceedings of the 21ST annual international symposium on Computer architecture ISCA '94**, Volume 22 Issue 2

Publisher: IEEE Computer Society Press, ACM Press


Full text available:  pdf(1.07 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Accurate branch prediction is critical to performance; mispredicted branches mean that ten's of cycles may be wasted in superscalar architectures. Architectures combining very effective


branch prediction mechanisms coupled with modified branch target buffers (BTB's) have been proposed for wide-issue processors. These mechanisms require considerable processor resources. Concurrently, the larger address space of 64-bit architectures introduce new obstacles and opportunities. A larger address space ...

10 Optimally profiling and tracing programs

 Thomas Ball, James R. Larus

 February 1992 **Proceedings of the 19th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**

Publisher: ACM Press

Full text available:  [pdf\(1.27 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


This paper presents algorithms for inserting monitoring code to profile and trace programs. These algorithms greatly reduce the cost of measuring programs. Profiling counts the number of times each basic block in a program executes and has a variety of applications. Instruction traces are the basis for trace-driven simulation and analysis, and are also used in trace-driven debugging. The profiling algorithm chooses a placement of counters that is optimized—and frequently op ...

11 Consistent overhead Byte stuffing

Stuart Cheshire, Mary Baker

April 1999 **IEEE/ACM Transactions on Networking (TON)**, Volume 7 Issue 2


Publisher: IEEE Press

Full text available:  [pdf\(263.90 KB\)](#)

Additional Information: [full citation](#), [references](#), [index terms](#)


Keywords: Byte stuffing, framing, packet, serial, transmission

12 Consistent overhead byte stuffing

 Stuart Cheshire, Mary Baker

 October 1997 **ACM SIGCOMM Computer Communication Review , Proceedings of the ACM SIGCOMM '97 conference on Applications, technologies, architectures, and protocols for computer communication SIGCOMM '97**, Volume 27 Issue 4

Publisher: ACM Press

Full text available:  [pdf\(1.76 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


Byte stuffing is a process that transforms a sequence of data bytes that may contain 'illegal' or 'reserved' values into a potentially longer sequence that contains no occurrences of those values. The extra length is referred to in this paper as the overhead of the algorithm. To date, byte stuffing algorithms, such as those used by SLIP [RFC1055], PPP [RFC1662] and AX.25 [ARRL84], have been designed to incur low average overhead but have made little effort to minimize worst case overhead. Some inc ...

13 RATCHET: real-time address trace compression hardware for extended traces

 Colleen D. Schieber, Eric E. Johnson

 April 1994 **ACM SIGMETRICS Performance Evaluation Review**, Volume 21 Issue 3-4

Publisher: ACM Press

Full text available:  [pdf\(783.24 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)


The address traces used in computer architecture research are commonly generated using software techniques that introduce time dilations of an order of magnitude or more. Such techniques may also omit classes of memory references that are important for accurate models of computer systems, such as instruction prefetches, operating system references, and interrupt activity. We describe a technique for capturing all classes of references in real time. RATCHET employs trace filtering hardware to redu ...

14 [SAVE: an algorithm for smoothed adaptive video over explicit rate networks](#)

N. G. Duffield, K. K. Ramakrishnan, Amy R. Reibman


December 1998 **IEEE/ACM Transactions on Networking (TON)**, Volume 6 Issue 6

Publisher: IEEE Press

Full text available:  [pdf\(539.13 KB\)](#) Additional Information: [full citation](#), [references](#), [citings](#), [index terms](#)


Keywords: compressed video, multiplexing, rate control, smoothing

15 [Instruction path coprocessors](#)

 Yuan Chou, John Paul Shen


May 2000 **ACM SIGARCH Computer Architecture News , Proceedings of the 27th annual international symposium on Computer architecture ISCA '00**, Volume 28 Issue 2

Publisher: ACM Press

Full text available:  [pdf\(134.64 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citings](#), [index terms](#)


This paper presents the concept of an Instruction Path Coprocessor (I-COP), which is a programmable on-chip coprocessor, with its own mini-instruction set, that operates on the core processor's instructions to transform them into an internal format that can be more efficiently executed. It is located off the critical path of the core processor to ensure that it does not negatively impact the core processor's cycle time or pipeline depth. An I-COP is highly versatile and can be used ...

16 [Incorporating speculative execution in exact control-dependent scheduling](#)


 Ivan Radivojevic, Forrest Brewer

June 1994 **Proceedings of the 31st annual conference on Design automation**

Publisher: ACM Press


Full text available:  [pdf\(73.18 KB\)](#) Additional Information: [full citation](#), [references](#), [citings](#), [index terms](#)

17 [Unconstrained speculative execution with predicated state buffering](#)

 Hideki Ando, Chikako Nakanishi, Tetsuya Hara, Masao Nakaya

May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture ISCA '95**, Volume 23 Issue 2

Publisher: ACM Press

Full text available:  pdf(1.50 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)



Speculative execution is execution of instructions before it is known whether these instructions should be executed. Compiler-based speculative execution has the potential to achieve both a high instruction per cycle rate and high clock rate. Pure compiler-based approaches, however, have greatly limited instruction scheduling due to a limited ability to handle side effects of speculative execution. Significant performance improvement is, thus, difficult in non-numerical applications. This paper ...

18 [Improving trace cache effectiveness with branch promotion and trace packing](#)

 Sanjay Jeram Patel, Marius Evers, Yale N. Patt

April 1998 **ACM SIGARCH Computer Architecture News , Proceedings of the 25th annual international symposium on Computer architecture ISCA '98**, Volume 26 Issue 3

Publisher: IEEE Computer Society, ACM Press

Full text available:  pdf(1.11 MB) 

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

[Publisher Site](#)


The increasing widths of superscalar processors are placing greater demands upon the fetch mechanism. The trace cache meets these demands by placing logically contiguous instructions in physically contiguous storage. As a result, the trace cache delivers instructions at a high rate by supplying multiple fetch blocks each cycle. In this paper, we examine two techniques to improve the number of instructions delivered each cycle by the trace cache. The first technique, branch promotion, dynamically ...

19 [An analysis of the information content of address reference streams](#)

 Jeffrey C. Becker, Arvin Park, Matthew Farrens

September 1991 **Proceedings of the 24th annual international symposium on Microarchitecture**

Publisher: ACM Press

Full text available:  pdf(618.07 KB)



Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

20 [Branch prediction based on universal data compression algorithms](#)

 Eitan Federovsky, Meir Feder, Sholomo Weiss

April 1998 **ACM SIGARCH Computer Architecture News , Proceedings of the 25th annual international symposium on Computer architecture ISCA '98**, Volume 26 Issue 3

Publisher: IEEE Computer Society, ACM Press

Full text available:  pdf(987.72 KB) 

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

[Publisher Site](#)

Data compression and prediction are closely related. Thus prediction methods based on data compression algorithms have been suggested for the branch prediction problem. In this work we consider two universal compression algorithms: prediction by partial matching (PPM), and a recently developed method, context tree weighting (CTW). We describe the prediction algorithms induced by these methods. We also suggest adaptive algorithms --- variations of the

basic methods that attempt to fit limited mem ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM
Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Playe](#)



USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Log out](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used **compress trace conditional partial**Found **1,858** of **169**Sort results
by
☒ [Save results to a Binder](#)
[Try an Advanced Search](#)
☐ [Search Tips](#)
[Try this search in The ACM Guide](#)
Display
results
☐ [Open results in a new window](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐

1 [Improving trace cache effectiveness with branch promotion and trace packing](#)



Sanjay Jeram Patel, Marius Evers, Yale N. Patt

 April 1998 **ACM SIGARCH Computer Architecture News , Proceedings of the 25th annual international symposium on Computer architecture ISCA '98**, Volume 26 Issue 3

Publisher: IEEE Computer Society, ACM Press
Full text available: [pdf\(1.11](#)
[MB\)](#)
[Publisher Site](#)

 Additional Information: [full citation](#), [abstract](#), [references](#),
[citations](#), [index terms](#)

The increasing widths of superscalar processors are placing greater demands upon the fetch mechanism. The trace cache meets these demands by placing logically contiguous instructions in physically contiguous storage. As a result, the trace cache delivers instructions at a high rate by supplying multiple fetch blocks each cycle. In this paper, we examine two techniques to improve the number of instructions delivered each cycle by the trace cache. The first technique, branch promotion, dynamically ...

2 [Branch prediction based on universal data compression algorithms](#)



Eitan Federovsky, Meir Feder, Sholomo Weiss

 April 1998 **ACM SIGARCH Computer Architecture News , Proceedings of the 25th annual international symposium on Computer architecture ISCA '98**, Volume 26 Issue 3


Publisher: IEEE Computer Society, ACM Press
Full text available: [pdf\(987.72](#)
[KB\)](#)
[Publisher Site](#)

 Additional Information: [full citation](#), [abstract](#), [references](#),
[citations](#), [index terms](#)

Data compression and prediction are closely related. Thus prediction methods based on data compression algorithms have been suggested for the branch prediction problem. In this work we consider two universal compression algorithms: prediction by partial matching (PPM), and a recently developed method, context tree weighting (CTW). We describe the prediction algorithms induced by these methods. We also suggest adaptive algorithms --- variations of the basic methods that attempt to fit limited mem ...


3

[GPGPU: general purpose computation on graphics hardware](#)

 David Luebke, Mark Harris, Jens Krüger, Tim Purcell, Naga Govindaraju, Ian Buck, Cliff Woolley, Aaron Lefohn

August 2004 **Proceedings of the conference on SIGGRAPH 2004 course notes GRAPH '04**


Publisher: ACM Press

Full text available:  pdf(63.03 MB)

Additional Information: [full citation](#), [abstract](#)


The graphics processor (GPU) on today's commodity video cards has evolved into an extremely powerful and flexible processor. The latest graphics architectures provide tremendous memory bandwidth and computational horsepower, with fully programmable vertex and pixel processing units that support vector operations up to full IEEE floating point precision. High level languages have emerged for graphics hardware, making this computational power accessible. Architecturally, GPUs are highly parallel s ...

4 Level set and PDE methods for computer graphics

 David Breen, Ron Fedkiw, Ken Museth, Stanley Osher, Guillermo Sapiro, Ross Whitaker

August 2004 **Proceedings of the conference on SIGGRAPH 2004 course notes GRAPH '04**

Publisher: ACM Press

Full text available:  pdf(17.07 MB)

Additional Information: [full citation](#), [abstract](#)


Level set methods, an important class of partial differential equation (PDE) methods, define dynamic surfaces implicitly as the level set (iso-surface) of a sampled, evolving nD function. The course begins with preparatory material that introduces the concept of using partial differential equations to solve problems in computer graphics, geometric modeling and computer vision. This will include the structure and behavior of several different types of differential equations, e.g. the level set eq ...

5 A framework for the integration of partial evaluation and abstract interpretation of logic programs

 Michael Leuschel

May 2004 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 26 Issue 3

Publisher: ACM Press

Full text available:  pdf(319.71 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Recently the relationship between abstract interpretation and program specialization has received a lot of scrutiny, and the need has been identified to extend program specialization techniques so as to make use of more refined abstract domains and operators. This article clarifies this relationship in the context of logic programming, by expressing program specialization in terms of abstract interpretation. Based on this, a novel specialization framework, along with generic correctness results ...



Keywords: Partial deduction, abstract interpretation, flow analysis, logic programming, partial evaluation, program transformation

6 Alternative fetch and issue policies for the trace cache fetch mechanism

Daniel Holmes Friendly, Sanjay Jeram Patel, Yale N. Patt

December 1997 **Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture**

Publisher: IEEE Computer Society


Full text available:  [pdf\(1.16 MB\)](#)
 [Publisher Site](#)

Additional Information: [full citation](#), [abstract](#), [references](#),
[citations](#), [index terms](#)

The increasing widths of superscalar processors are placing greater demands upon the fetch mechanism. The trace cache meets these demands by placing logically contiguous instructions in physically contiguous storage. It is capable of supplying multiple fetch blocks each cycle. In this paper we examine two fetch and issue techniques, partial matching and inactive issue, that improve the overall performance of the trace cache by improving the effective fetch rate. We show that for the SPECint95 be ...


Keywords: high bandwidth fetch mechanisms, trace cache, wide issue machines, speculative execution, partial matching, inactive issue

7 [The elements of nature: interactive and realistic techniques](#)

 Oliver Deussen, David S. Ebert, Ron Fedkiw, F. Kenton Musgrave, Przemyslaw Prusinkiewicz, Doug Roble, Jos Stam, Jerry Tessendorf

August 2004 **Proceedings of the conference on SIGGRAPH 2004 course notes GRAPH '04**

Publisher: ACM Press

Full text available:  [pdf\(17.65 MB\)](#)

Additional Information: [full citation](#), [abstract](#)


This updated course on simulating natural phenomena will cover the latest research and production techniques for simulating most of the elements of nature. The presenters will provide movie production, interactive simulation, and research perspectives on the difficult task of photorealistic modeling, rendering, and animation of natural phenomena. The course offers a nice balance of the latest interactive graphics hardware-based simulation techniques and the latest physics-based simulation techni ...

8 [Completeness and incompleteness of trace-based network proof systems](#)

 J. Widom, D. Gries, F. B. Schneider

October 1987 **Proceedings of the 14th ACM SIGACT-SIGPLAN symposium on Principles of programming languages**

Publisher: ACM Press

Full text available:  [pdf\(1.30 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#),
[citations](#), [index terms](#)

Most trace-based proof systems for networks of processes are known to be incomplete. Extensions to achieve completeness are generally complicated and cumbersome. In this paper, a simple trace logic is defined and two examples are presented to show its inherent incompleteness. Surprisingly, both examples consist of only one process, indicating that network composition is not a cause of incompleteness. Axioms necessary and sufficient for the relative completeness of a trace logic are then pre ...

9 [Fast detection of communication patterns in distributed executions](#)

Thomas Kunz, Michiel F. H. Seuren


November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Publisher: IBM Press

Full text available:  [pdf\(4.21 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

10 Partial method compilation using dynamic profile information

 John Whaley


October 2001 **ACM SIGPLAN Notices , Proceedings of the 16th ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications OOPSLA '01**, Volume 36 Issue 11

Publisher: ACM Press

Full text available:  [pdf\(1.73 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


The traditional tradeoff when performing dynamic compilation is that of fast compilation time versus fast code performance. Most dynamic compilation systems for Java perform selective compilation and/or optimization at a method granularity. This is not the optimal granularity level. However, compiling at a sub-method granularity is thought to be too complicated to be practical. This paper describes a straightforward technique for performing compilation and optimizations at a finer, sub-metho ...

11 Modeling and simulation of self-similar variable bit rate compressed video: a unified approach

 Changcheng Huang, Michael Devetsikiotis, Ioannis Lambadaris, A. Roger Kaye


October 1995 **ACM SIGCOMM Computer Communication Review , Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication SIGCOMM '95**, Volume 25 Issue 4

Publisher: ACM Press

Full text available:  [pdf\(1.06 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


Variable bit rate (VBR) compressed video is expected to become one of the major loading factors in high-speed packet networks such as ATM-based B-ISDN. However, recent measurements based on long empirical traces (complete movies) revealed that VBR video traffic possesses *self-similar* (or *fractal*) characteristics, meaning that the dependence in the traffic stream lasts much longer than traditional models can capture. In this paper, we present a unified approach which, in addition to ...

12 Ordinal Hierarchies and Naming Complexity Classes


 Leonard Bass, Paul Young


October 1973 **Journal of the ACM (JACM)**, Volume 20 Issue 4

Publisher: ACM Press

Full text available:  [pdf\(1.46 MB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)


13 A comparison of full and partial predicated execution support for ILP processors

-  Scott A. Mahlke, Richard E. Hank, James E. McCormick, David I. August, Wen-Mei W. Hwu
 May 1995 **ACM SIGARCH Computer Architecture News , Proceedings of the 22nd annual international symposium on Computer architecture ISCA '95**, Volume 23 Issue 2
Publisher: ACM Press


Full text available:  [pdf\(1.48 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#),
[citings](#), [index terms](#)

One can effectively utilize predicated execution to improve branch handling in instruction-level parallel processors. Although the potential benefits of predicated execution are high, the tradeoffs involved in the design of an instruction set to support predicated execution can be difficult. On one end of the design spectrum, architectural support for full predicated execution requires increasing the number of source operands for all instructions. Full predicate support provides for the most fle ...

14 Parallelizing nonnumerical code with selective scheduling and software pipelining

-  Soo-Mook Moon, Kemal Ebcioglu
 November 1997 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,
 Volume 19 Issue 6

Publisher: ACM Press

Full text available:  [pdf\(543.93 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#),
[citings](#), [index terms](#)


Instruction-level parallelism (ILP) in nonnumerical code is regarded as scarce and hard to exploit due to its irregularity. In this article, we introduce a new code-scheduling technique for irregular ILP called "selective scheduling" which can be used as a component for superscalar and VLIW compilers. Selective scheduling can compute a wide set of independent operations across all execution paths based on renaming and forward-substitution and can compute availab ...

Keywords: VLIW, global instruction scheduling, instruction-level parallelism, software pipelining, speculative code motion, superscalar

15 Static correlated branch prediction

-  Cliff Young, Michael D. Smith
 September 1999 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,
 Volume 21 Issue 5



Publisher: ACM Press

Full text available:  [pdf\(508.49 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#),
[citings](#), [index terms](#)



Recent work in history-based branch prediction uses novel hardware structures to capture branch correlation and increase branch prediction accuracy. Branch correlation occurs when the outcome of a conditional branch can be accurately predicted by observing the outcomes of previously executed branches in the dynamic instruction stream. In this article, we show how to instrument a program so that it is practical to collect run-time statistics that indicate where branch correl ...

Keywords: branch correlation, branch prediction, path profiling, profile-driven optimization



16 Formalizing and integrating the dynamic model within OMT

-  Enoch Y. Wang, Heather A. Richter, Betty H. C. Cheng
 May 1997 **Proceedings of the 19th international conference on Software engineering**
Publisher: ACM Press
 Full text available:  pdf(2.01 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: design, formal specification, model integration, object-oriented modeling, requirements analysis


- 17 [XML indexing and compression: XPRESS: a queriable compression for XML data](#)
 Jun-Ki Min, Myung-Jae Park, Chin-Wan Chung
 June 2003 **Proceedings of the 2003 ACM SIGMOD international conference on Management of data**
Publisher: ACM Press
 Full text available:  pdf(277.17 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Like HTML, many XML documents are resident on native file systems. Since XML data is irregular and verbose, the disk space and the network bandwidth are wasted. To overcome the verbosity problem, the research on compressors for XML data has been conducted. However, some XML compressors do not support querying compressed data, while other XML compressors which support querying compressed data blindly encode tags and data values using predefined encoding methods. Thus, the query performance on com ...


- 18 [Relational queries over program traces](#)
 Simon Goldsmith, Robert O'Callahan, Alex Aiken
 October 2005 **ACM SIGPLAN Notices , Proceedings of the 20th annual ACM SIGPLAN conference on Object oriented programming systems languages and applications OOPSLA '05, Volume 40 Issue 10**
Publisher: ACM Press
 Full text available:  pdf(192.17 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Instrumenting programs with code to monitor runtime behavior is a common technique for profiling and debugging. In practice, instrumentation is either inserted manually by programmers, or automatically by specialized tools that monitor particular properties. We propose Program Trace Query Language (PTQL), a language based on relational queries over program traces, in which programmers can write expressive, declarative queries about program behavior. We also describe our compiler, Partique

Keywords: PTQL, partique, program trace query language, relational

- 19 [Analysis of branch prediction via data compression](#)
 I-Cheng K. Chen, John T. Coffey, Trevor N. Mudge
 September 1996 **ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , Proceedings of the seventh international conference on Architectural support for programming languages and operating systems ASPLOS-VII, Volume 31 , 30 Issue 9 , 5**

Publisher: ACM Press

Full text available:  pdf(930.57 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


Branch prediction is an important mechanism in modern microprocessor design. The focus of research in this area has been on designing new branch prediction schemes. In contrast, very few studies address the theoretical basis behind these prediction schemes. Knowing this theoretical basis helps us to evaluate how good a prediction scheme is and how much we can expect to improve its accuracy. In this paper, we apply techniques from data compression to establish a theoretical basis for branch predic ...

20 [Optimal tracing and incremental reexecution for debugging long-running programs](#)

 Robert H. B. Netzer, Mark H. Weaver

June 1994 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1994 conference on Programming language design and implementation PLDI '94**, Volume 29 Issue 6

Publisher: ACM Press


Full text available:  pdf(1.34 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Playe](#)



[Home](#) | [Login](#) | [Logout](#) | [Access Information](#)
[Site](#)

Welcome United States Patent and Trademark
Office

Search Results

[BROWSE](#)

[SEARCH](#)

[IEEE XPLORE
GUIDE](#)

Results for "(((compress* uncompress* trace branch start)<in>metadata)) <and>

(pyr >= 1980 <an..."

Your search matched 0 documents.

A maximum of 100 results are displayed, 25 to a page, sorted by **Relevance** in
Descending order.

e-mail

» Search Options

[View Session History](#)

[New Search](#)

[Modify Search](#)

(((compress* uncompress* trace branch start)<in>metadata)) <and> (pyr

☐ Check to search only within this results set

» Key

IEEE
JNL

IEEE Journal or
Magazine

IEE
JNL

IEE Journal or
Magazine

IEEE
CNF

IEEE Conference
Proceeding

IEE
CNF

IEE Conference
Proceeding

IEEE
STD

IEEE Standard

Display
Format:

☒ Citation ☐ Citation & Abstract

No results were found.

Please edit your search criteria and try again. Refer to the Help pages
assistance revising your search.

[Help](#) [Contact Us](#)
[Security](#)

© Copyright 2006
Ri

Indexed by
 Inspec®



[Home](#) | [Login](#) | [Logout](#) | [Access Information](#)
[Site](#)

Welcome United States Patent and Trademark
Office

Search Results

[BROWSE](#)

[SEARCH](#)

[IEEE XPLORE
GUIDE](#)

Results for "(((compress* uncompress* trace* branch* start*)<in>metadata))

<and> (pyr >= 1980 <..."

Your search matched 0 documents.

A maximum of 100 results are displayed, 25 to a page, sorted by **Relevance** in
Descending order.

e-mail

» Search Options

[View Session History](#)

[New Search](#)

[Modify Search](#)

☐ Check to search only within this results set

» Key

IEEE
JNL

IEEE Journal or
Magazine

IEE
JNL

IEE Journal or
Magazine

IEEE
CNF

IEEE Conference
Proceeding

IEE
CNF

IEE Conference
Proceeding

IEEE
STD

IEEE Standard

Display
Format:

☒ Citation ☐ Citation & Abstract

No results were found.

Please edit your search criteria and try again. Refer to the Help pages
assistance revising your search.

[Help](#) [Contact Us](#)
[Security](#)

© Copyright 2006
 IEEE

Indexed by
 Inspec



[Home](#) | [Login](#) | [Logout](#) | [Access Information](#)
[Site](#)

Welcome United States Patent and Trademark
Office

Search Results

[BROWSE](#)

[SEARCH](#)

[IEEE XPLORE
GUIDE](#)

Results for "(((compress* trace*)<in>metadata)) <and> (pyr >= 1980 <and> pyr <= 2001))" [e-mail](#)

Your search matched 1 of 1306777 documents.

A maximum of 100 results are displayed, 25 to a page, sorted by **Relevance** in **Descending** order.

» Search Options

[View Session History](#)

[New Search](#)

[Modify Search](#)

(((compress* trace*)<in>metadata)) <and> (pyr >= 1980 <and> pyr <= 2001))

☐ Check to search only within this results set

» Key

IEEE JNL IEEE Journal or Magazine

IEE JNL IEE Journal or Magazine

IEEE CNF IEEE Conference Proceeding

IEE CNF IEE Conference Proceeding

IEEE STD IEEE Standard

[Display
Format:](#)

☒ Citation ☐ Citation & Abstract

☐ 1. Method of correction for linear distortion of signals and to DNA fingerprint data

Menacer, M.; Woolfson, M.S.; Crowe, J.A.;
Electronics Letters

Volume 28, Issue 23, 5 Nov. 1992 Page(s):2126 - 2127

[AbstractPlus](#) | [Full Text: PDF\(168 KB\)](#) **IEE JNL**

[Help](#) [Contact Us](#)
[Security](#)

Indexed by
Inspec

© Copyright 2006
Ri